IN THE CLAIMS

1. (Currently Amended) A method of operating a memory device driver comprising:

    querying at least one memory device to discover the memory type by reading a memory ID code stored in the memory device; and

    configuring the driver to access the at least one memory device according to the discovered memory type by matching the discovered memory type to an entry in a table and loading parameters from the table entry that matches the discovered memory type;

    wherein the table the table is not stored on the at least one memory device and is modifiable to update the entries of memory types and parameters.

2. (previously presented) The method of claim 1, wherein querying at least one memory device to discover the memory type by reading a memory ID code stored in the memory device further comprises querying at least one Flash memory device to discover the memory type.

3. (original) The method of claim 2, wherein the at least one Flash memory device is one of a NOR Flash and a NAND Flash.

4. (original) The method of claim 2, wherein querying at least one memory device to discover the memory type further comprises writing to an address of the memory device and reading a response from the address.

5. (original) The method of claim 2, wherein querying at least one memory device to discover the memory type further comprises querying a common Flash interface (CFI) of the memory device.

6. (original) The method of claim 2, wherein querying at least one memory device to discover the memory type further comprises querying a protection register of the memory device.

7.   (previously presented) The method of claim 1, wherein querying at least one memory device to discover the memory type by reading a memory ID code stored in the memory device further comprises querying an addressable memory ID stored in the memory device.

8.   (previously presented) The method of claim 1, wherein querying at least one memory device to discover the memory type by reading a memory ID code stored in the memory device further comprises querying an architecture feature of the memory device.

9.   (original) The method of claim 1, wherein the driver contains at least one of a low level driver, a data manager, and a file manager.

10.   (Currently Amended) The method of claim 1, wherein configuring the driver to access the at least one memory device according to the discovered memory type <u>by matching the discovered memory type to an entry in a table and loading parameters from the table entry that matches the discovered memory type</u> further comprises configuring the driver to access the at least one memory device with low level driver operation parameters and memory device command sequences to match the discovered memory type.

11.   (Currently Amended) A method of operating a memory device driver comprising: querying at least one memory device to discover the memory type; and configuring the driver to access the at least one memory device according to the discovered memory type;
        The method of claim 1, wherein configuring the driver to access the at least one memory device according to the discovered memory type <u>by matching the discovered memory type to an entry in a table and loading parameters from the table entry that matches the discovered memory type</u> further comprises configuring the driver to access the at least one memory device with data manager parameters and procedures to match the discovered memory type.

12.   (Currently Amended) ~~A method of operating a memory device driver comprising:~~

~~querying at least one memory device to discover the memory type; and~~

~~configuring the driver to access the at least one memory device according to the~~

~~discovered memory type;~~

The method of claim 1, wherein configuring the driver to access the at least one memory

device according to the discovered memory type by matching the discovered memory

type to an entry in a table and loading parameters from the table entry that matches

the discovered memory type further comprises configuring the driver to access the at

least one memory device with file manager parameters and procedures to match the

discovered memory type.

13.   (Cancelled)

14.   (Currently Amended) The method of claim 1 ~~claim 13~~, wherein the table contains at least

one of a low level driver configuration, a data manager configuration, and a file manager

configuration.

15.   (Currently Amended) The method of claim 1 ~~claim 13~~, wherein the table contains one or

more entries for a selected memory type, where each of the one or more entries contains a

different access data model.

16.   (Cancelled)

17.   (Currently Amended) A method of operating a system comprising:

querying at least one Flash memory device coupled to a host to discover the memory type

by reading a device ID and/or manufacturer code stored in the Flash memory device;

and

configuring a driver routine executing on the host to access the at least one Flash memory

device according to the discovered memory type <u>by loading parameters from an entry</u>

<u>in a table that matches the discovered memory type;</u>

<u>wherein the table is not stored on the at least one Flash memory device and is modifiable</u>

<u>to update the entries of memory types and parameters.</u>

18.   (original) The method of claim 17, wherein the at least one Flash memory device is one

of a NOR Flash and a NAND Flash.

19.   (original) The method of claim 17, wherein an interface to the Flash memory device is

one of a PCMCIA-ATA, a Compact Flash (CF), a USB Flash, MemoryStick, and a

multimedia card (MMC) compatible interface.

20.   (previously presented) The method of claim 17, wherein querying at least one Flash

memory device coupled to a host to discover the memory type by reading a device ID

and/or manufacturer code stored in the Flash memory device further comprises querying

a protection register of the Flash memory device.

21.   (Currently Amended) The method of claim 17, <u>wherein</u> querying at least one Flash

memory device coupled to a host to discover the memory type by reading a device ID

and/or manufacturer code stored in the Flash memory device further comprises querying

a memory ID stored in the Flash memory device.

22.   (previously presented) The method of claim 17, wherein querying at least one Flash

memory device coupled to a host to discover the memory type by reading a device ID

and/or manufacturer code stored in the Flash memory device further comprises querying

a common Flash interface (CFI) of the Flash memory device.

23.   (previously presented) The method of claim 17, wherein querying at least one Flash

memory device coupled to a host to discover the memory type by reading a device ID

and/or manufacturer code stored in the Flash memory device further comprises issuing a

Flash command to an address and reading the result from the address.

24.   (Currently Amended) The method of claim 17, wherein configuring a driver routine

executing on the host to access the at least one Flash memory device according to the

discovered memory type <u>by loading parameters from an entry in a table that matches the</u>

<u>discovered memory type</u> further comprises configuring the driver routine to access the at

least one Flash memory device with operation parameters and memory device command

sequences to match the discovered memory type.

25.   (Currently Amended) ~~A method of operating a system comprising:~~

~~querying at least one Flash memory device coupled to a host to discover the memory~~

~~type;~~

~~configuring a driver routine executing on the host to access the at least one Flash memory~~

~~device according to the discovered memory type; and~~

<u>The method of claim 17, configuring a driver routine executing on the host to access the</u>

<u>at least one Flash memory device according to the discovered memory type by</u>

<u>loading parameters from an entry in a table that matches the discovered memory type</u>

<u>further comprises</u> configuring the driver routine to access the at least one Flash

memory device with memory management routines to match the discovered memory

type.

26.   (Cancelled)

27.   (Cancelled)

28.   (original) The method of claim 17, wherein the host is one of a processor and an external

memory controller.

29.   (Currently Amended) A method of configuring a driver comprising:

querying at least one Flash memory device ~~to read a device ID and/or manufacturer code stored in the Flash memory device~~ to discover the memory type; and

configuring the driver to access the at least one memory device according to the discovered memory type <u>by matching the discovered memory type to an entry in a table and loading parameters from the table entry that matches the discovered memory type, wherein the table is not stored on the at least one Flash memory device and where the table is updateable</u>.

30.   (original) The method of claim 29, wherein the at least one Flash memory device is one of a NOR Flash and a NAND Flash.

31.   (original) The method of claim 29, wherein an interface to the Flash memory device is one of a PCMCIA-ATA, a Compact Flash (CF), a USB Flash, MemoryStick, and a multimedia card (MMC) compatible interface.

32.   (Currently Amended) The method of claim 29, wherein querying at least one Flash memory device ~~to read a device ID and/or manufacturer code stored in the Flash memory device~~ to discover the memory type further comprises querying a protection register of the Flash memory device.

33.   (Currently Amended) The method of claim 29, wherein querying at least one Flash memory device ~~to read a device ID and/or manufacturer code stored in the Flash memory device~~ to discover the memory type further comprises querying a common Flash interface (CFI) of the Flash memory device.

34.   (Currently Amended) The method of claim 29, wherein querying at least one Flash memory device ~~to read a device ID and/or manufacturer code stored in the Flash memory device~~ to discover the memory type further comprises writing to an address of the Flash memory device and reading a result on the address.

35.   (original) The method of claim 29, wherein the driver contains at least one of a low level
        driver, a data manager, and a file manager.

36.   (Currently Amended) <u>The method of claim 29, wherein querying at least one Flash
        memory device to discover the memory type further comprises querying at least one
        Flash memory device to discover the memory type to read a device ID and/or
        manufacturer code stored in the Flash memory device</u> A method of configuring a driver
        comprising:
        querying at least one Flash memory device to discover the memory type;
        configuring the driver to access the at least one memory device according to the
            discovered memory type; and
        configuring the driver to access the at least one Flash memory device by matching the
            discovered memory type to an entry in a table and loading parameters from the table
            entry that matches the discovered memory type.

37.   (Currently Amended) The method of <u>claim 29</u> claim 36, wherein the table contains at
        least one of a low level driver configuration, a data manager configuration, and a file
        manager configuration.

38.   (Currently Amended) The method of <u>claim 29</u> claim 36, wherein the table contains one or
        more entries for a selected memory type, where each of the one or more entries contains a
        different access data model.

39.   (Currently Amended) The method of <u>claim 29</u> claim 36, wherein an access data model
        for the at least one Flash memory is selected from the table.

40.   (Currently Amended) A system comprising:
        at least one Flash memory device; and
        a host coupled to the at least one Flash memory device, wherein the host is adapted to
            query the at least one Flash memory device to read a device ID and/or manufacturer

code stored in the Flash memory device and configure a driver routine to access the at

least one Flash memory device in response to the query <u>by matching the discovered</u>

<u>device ID and/or manufacturer code to an entry in a table and loading parameters</u>

<u>from the table entry that matches the device ID and/or manufacturer code;</u>

<u>wherein the table is not stored on the at least one Flash memory device and is modifiable</u>

<u>to update the entries.</u>

41.   (original) The system of claim 40, wherein the at least one Flash memory device is one of
a NAND Flash and a NOR Flash.

42.   (original) The system of claim 40, wherein an interface to the Flash memory device is
one of a PCMCIA-ATA, a Compact Flash (CF), a USB Flash, MemoryStick, and a
multimedia card (MMC) compatible interface.

43.   (original) The system of claim 40, wherein the host is one of a processor and an external
memory controller.

44.   (original) The system of claim 40, wherein the host is adapted to query a protection
register of the Flash memory device.

45.   (previously presented) The system of claim 40, wherein the host is adapted to query a
memory ID stored at a specific address in the Flash memory device.

46.   (original) The system of claim 40, wherein the host is adapted to query a common Flash
interface (CFI) of the Flash memory device.

47.   (original) The system of claim 40, wherein the host is adapted to query an address of the
Flash memory device.

48.   (Currently Amended) A machine-usable medium, the machine-usable medium containing a software routine for causing a processor to execute a method, wherein the method comprises:

querying at least one Flash memory device to read a device ID and/or manufacturer code stored in the Flash memory device to discover the memory type; and

configuring a driver to access the at least one Flash memory device according to the discovered memory type by matching the discovered device ID and/or manufacturer code to an entry in a table and loading parameters from the table entry that matches the device ID and/or manufacturer code;

wherein the table is not stored on the at least one Flash memory device and is modifiable to update the entries.

49.   (original) The machine-usable medium of claim 48, wherein an interface to the at least one Flash memory device is one of a PCMCIA-ATA, a Compact Flash (CF), a USB Flash, MemoryStick, and a multimedia card (MMC) compatible interface.

50.   (previously presented) The machine-usable medium of claim 48, wherein querying at least one memory device to read a device ID and/or manufacturer code stored in the Flash memory device to discover the memory type further comprises querying a protection register of the Flash memory device.

51.   (previously presented) The machine-usable medium of claim 48, wherein querying at least one memory device to read a device ID and/or manufacturer code stored in the Flash memory device to discover the memory type further comprises querying a memory ID stored in the array of the Flash memory device.

52.   (previously presented) The machine-usable medium of claim 48, wherein querying at least one memory device to read a device ID and/or manufacturer code stored in the Flash memory device to discover the memory type further comprises querying a common Flash interface (CFI) of the Flash memory device.

53.   (Currently Amended) A system comprising:

at least one Flash memory device; and

a host coupled to the at least one Flash memory device, wherein the host comprises a

means for detecting a Flash memory type of the at least one Flash memory device by

reading a memory ID code stored in the Flash memory device and comprises a means

for configuring a driver to access the at least one Flash memory device in response to

the Flash memory type detected by the means for detecting by matching the memory

ID code to an entry in an data storage means and loading parameters from the entry;

wherein the data storage means is not stored on the at least one Flash memory device and

is modifiable to update the entries.